

Andrew Walters  
Joshua Steward  
Lowell Batacan  
Jessica Perez  
Zane Gittins  
Zachery Van Es

## Software Process

For our software process we intend to use an incremental approach to develop our software.

- Week 1: Preparing Interfaces and Code Base for Implementation. Learning Systems Requirements.  
Software specification:
  - This will be a very general setup for how the game mechanics will work; base player models, base AI and pathfinding frameworks in place, learning about unity frameworks and coding practices, setting up the github repository with settings built for Unity and C# code sharing.Implementation:
  - Each person works on specific modules in design process after base knowledge base is developed
  - Studying Unity API and tutorials
  - Solidifying how commit practices and specific branching will work in implementation stages
  - Solidifying knowledge in Unity Editor 2D world building and interface elementsValidation :
  - Testing code and editor practices with Visual Studio built in IDE Debugger and Debug tools in Unity Editor
  - Testing Editor Play Mode in Unity Editor
    - This will allow us to test each individual feature as the project progresses toward completion.
- Week 2: Base Implementations of AI systems, movement systems, world building progress.  
Software Specification:
  - Code base is received, implemented as specified for AI pathfinding and movementImplementation and Development:
  - Modifying code base for copying of public variables, data structures.
  - World Builder status: Base city layout with different types of structures for pathfinding and destinations laid out.
  - Built-in clock base code implemented for day-night systems.Validation:
  - All code compiles at all states.

- User is able to move around and camera follows the user.
- Play Mode yields positive results - we are able to play the game, with a player moving around and interacting through collision detects and raycasting with the environment.
- Week 3: AI Queue Based Destination Systems, advanced movement, detailed city layout with layers.

#### Software Specification:

- Specifications for this week will be to have a Queue of destinations for each individual AI to be able to follow, and a system to specify how these destinations are generated, scripts to add on to these destinations to allow for pathfinding, and logical cycles of destination adding based on player movements and player-AI interaction. Base AI prefabs should be created and added to.
- Implementation and Development - Code systems will require priority based queues, likely imported from pre-existing C# libraries. Pathfinding using the A\* pathfinding algorithm, used widely in many gaming applications, in contrast to Dijkstra's pathfinding algorithm which has slightly more overhead when dealing with many infinitesimal points in a user space.
- Validation - will test pathfinding on multiple AI, setup for implementing city wide AI system.
- Week 4: User level feedback on city wide AI transportation and movement

#### Software Specification:

- Visual elements of seeing system wide AI movement in logical pathfinding solutions are clear and precise. Bug testing and visual movement will commence on how AI elements work.
- Implementation and Development - will use built in user interfaces and prefabs that have been incrementally developed to develop city wide population simulator.
- Validation - when code base is created and tested, we should be able to enter play mode and witness player and AI models moving about the city in a logical and destination based pathfinding solution.
- Week 5: Virus and Phase Check Commencement

#### Software Specification:

- At this point, the system for AI moving about is created. We will now create systems for spreading the virus. This will involve multiple states on AI characters, multiple states of consciousness on player character, all represented visually so the player knows how the game is progressing, and score counters/clock counters to represent how the player is progressing in the game. This will be the initial challenge for how the player actually enjoys the game itself.

#### Implementation and Development:

- Will implement virus progression through mathematical lerp functions. This involves the use of frame by frame update in C# or javascript to update some current value on its previous value up to an end condition.

On end condition, terminates. Implemented recursively usually. End functions will be the AI characters dying from the virus

- Will implement systems for the virus to be spread by the player; interaction.

Validation: Will be able to spread effective virus throughout city population

- Week 6: Visual Improvements and Public transportation

Software Specification:

- Should have more visually appealing sprite characters and code design by the end of this segment
- Public transportation and public destinations set-up so player strategies can be implemented.

Implementation and Development:

- Will use built in C# and Unity functions to code visually appealing elements into the game. Graphics scripting and level coding involvement.
- Public transportation will be using modifiable structures to add to memory allocation.

Validation:

- We should have a fully playable game with a spreadable virus, public transportation and end goal of spreading the virus to the whole city population in a visually appealing manner as fast as possible. Playable in buildable model on PC and Mac.

Week2:

Software specification: Revise and improve upon general ideas